Adaptive Split Federated Learning for Network-Driven Cut Selection in Vehicular Edge Intelligence

Youngjoon Yang
Computer Science and Engineering
Kyungpook National University
Daegu, Korea
youngj719@knu.ac.kr

Sunghyun Kim
Computer Science and Engineering
Kyungpook National University
Daegu, Korea
sunghyunkim@knu.ac.kr

Chaehyeon Kim
Computer Science and Engineering
Kyungpook National University
Daegu, Korea
hyeon917@knu.ac.kr

Dongkyun Kim
Computer Science and Engineering
Kyungpook National University
Daegu, Korea
dongkyun@knu.ac.kr

Abstract*— Vehicular Edge Intelligence (VEI) enables Roadside Units (RSUs) to locally process roadway images, train object-recognition models, and trigger collision-warning functions. RSUs placed in different road environments observe different scene distributions, which leads to Non-Independent and Identically Distributed (Non-IID) local datasets. As a result, each RSU tends to overfit frequently observed classes while learning limited representations for rare but safety-critical objects, and this imbalance makes global convergence in distributed training unstable. To overcome the limitations of pure local learning, RSU–server collaborative training has been investigated as a more suitable approach, and Split Federated Learning (SFL) provides a practical architecture by combining split execution with parameter aggregation. However, vehicular communication channels vary over time in terms of bandwidth, distance, and interference, so a fixed cut layer in SFL cannot balance activation transmission cost and client-side feature expressiveness across RSUs. This paper proposes a channel-driven Adaptive Split Federated Learning (ASFL) scheme that selects the split layer dynamically according to real-time RSU–server link quality and available bandwidth. RSUs with weak channels are assigned shallower cuts, whereas RSUs with favorable links use deeper cuts to exploit richer feature extraction. Comparative experiments under different adaptive cut-selection configurations demonstrate that the proposed ASFL framework achieves improved model accuracy and reduced training latency by effectively balancing computation and communication overheads. These results indicate that network-aware adaptive partitioning is a promising strategy for distributed learning in VEI settings.

***Keywords*—Adaptive Split Learning, Vehicular Edge Intelligence, Non-IID, Channel-Driven Distributed Edge Learning**

I. Introduction

Intelligent Transportation Systems (ITS) integrate roadside sensing, V2X connectivity, and edge–cloud computing to enable real-time traffic monitoring and safety-critical decision making[1]. However, as large deployments of cameras, RSUs, and connected vehicles generate increasingly massive data streams, centralized cloud processing alone cannot satisfy the latency, bandwidth, and privacy requirements of modern transportation services. To address these limitations, Vehicular Edge Computing (VEC) places computation at RSUs close to where data is produced, reducing backbone traffic and improving response time[2]. Building on this paradigm, VEI supports object detection, traffic-scene understanding, and hazard prediction directly at the edge, providing real-time warnings and assisting vehicles in safety-critical decisions. By shortening the perception–decision loop and reducing dependence on remote cloud servers, VEI has become a core enabler of next-generation ITS applications that demand low-latency perception and reliable situational awareness across diverse road environments.

In VEI systems, RSUs equipped with camera-based edge devices operate as localized perception nodes that continuously collect and process roadway imagery. These edge devices execute real-time object-recognition pipelines that identify surrounding vehicles and vulnerable road users, track their trajectories, and estimate potential collision risks based on spatial and temporal cues. The extracted features and semantic information enable timely safety alerts, supporting vehicles in avoiding hazardous situations and improving overall roadway awareness[3]. By performing these tasks directly at the network edge, RSUs reduce reliance on remote cloud servers and help maintain low-latency responses essential for safety-critical ITS applications.

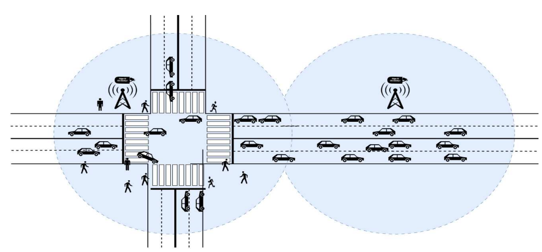

Fig 1. Non-IID data distributions between intersection and straight-road RSUs in a VEI environment

As shown in Fig. 1, RSUs deployed in different road environments observe traffic scenes with varying levels of complexity. Intersection RSUs encounter diverse road users—including pedestrians, bicycles, and vehicles approaching from multiple directions—whereas straight-road RSUs primarily observe uniform vehicle flows. These heterogeneous sensing conditions lead to Non-Independent and Identically Distributed (Non-IID) local datasets, causing each RSU to optimize its model toward the object classes most frequently observed in its environment. Such imbalance results in biased local updates and limited generalization capability when relying solely on local training. In safety-critical applications, this implies that an RSU may fail to detect objects that rarely appear in its own scene, despite their importance for overall system reliability[4]. These limitations highlight the need for collaborative training frameworks that can aggregate knowledge across RSUs and mitigate environment-specific learning biases.

Federated Learning (FL) allows RSUs to collaboratively train a global model without sharing raw data, thereby improving privacy and reducing backbone traffic[5]. However, because each RSU must train the entire model locally, FL imposes substantial computational load on resource-constrained edge devices, resulting in long local training latency and scalability limitations[6]. Split Learning (SL) addresses this issue by dividing the model at a predefined cut layer so that RSUs execute only the shallow portion and offload deeper layers to the server. This reduces device-side computation, but SL causes catastrophic forgetting due to its strict client-server sequential execution. This refers to the phenomenon where the model forgets data learned early in training[7]. To combine the advantages of both approaches, Split Federated Learning (SFL) integrates federated parameter aggregation with SL-style model partitioning, enabling parallel training across RSUs while reducing on-device computational burden. This hybrid architecture improves scalability and provides a more practical distributed learning framework for vehicular edge environments.

Despite its advantages, SFL inherits a structural limitation arising from the use of a fixed cut layer, which determines how computation and communication are divided between RSUs and the server. When the split position is shallow, RSUs offload most of the computation but must transmit large activation maps, increasing uplink overhead; conversely, deeper splits reduce communication volume but impose heavier on-device computation, creating an unfavorable trade-off that varies across RSUs[8]. This limitation becomes particularly problematic in vehicular environments, where communication quality fluctuates due to varying bandwidth availability, RSU–server distance, interference, and rapid changes in wireless channel conditions[9]. Under such dynamic conditions, a single static partition cannot balance computational load and communication cost for all RSUs, often leading to performance degradation and unstable training behavior. These challenges indicate the need for a more flexible learning framework capable of adapting the split position to real-time network conditions.

To overcome the inflexibility of fixed-cut SFL under fluctuating vehicular channels, this paper introduces a channel-driven Adaptive Split Federated Learning (ASFL) framework that adjusts the model partition dynamically for each RSU. The key idea is to ensure that computation and communication are balanced according to real-time link conditions, rather than enforcing the same split position across heterogeneous RSUs. By assuming comparable processing capability among RSUs and focusing exclusively on network-driven adaptation, ASFL avoids the limitations of approaches that rely on device-specific computation delay. Through adaptive selection of the cut layer—based on bandwidth availability, channel quality, and RSU–server link distance—ASFL reduces transmission overhead for RSUs experiencing weak channels while enabling richer feature extraction when the channel is strong. This flexibility addresses the instability and inefficiency inherent in fixed-cut SFL and provides a more robust foundation for collaborative training in dynamic vehicular environments.

The main contributions of this work are as follows:

- We propose a channel-driven ASFL framework that aims to dynamically adjust the model's cut layer for each RSU according to real-time network conditions.
- We formulate the split-selection problem from a network-centric perspective, highlighting how bandwidth, channel quality, and RSU–server distance influence activation size, computational load, and training stability.
- We design an adaptive cut-selection mechanism intended to balance communication overhead and client-side feature expressiveness under fluctuating vehicular channel conditions.
- We aim to enhance the robustness of distributed learning in heterogeneous VEI environments by enabling flexible partitioning rather than relying on a single fixed split position.
- We construct a realistic VEI evaluation scenario reflecting intersection and straight-road deployments to examine the behavior of RSUs with differing sensing characteristics.
- We perform comprehensive comparisons with FL, SL, SFL, and fixed-cut baselines to demonstrate the potential advantages of ASFL in terms of accuracy, convergence behavior, and training latency.

The remainder of this paper is structured as follows. Section II reviews the related works. Section III describes the system architecture and the RSU–server learning model. Section IV presents the proposed ASFL framework and the channel-driven split-selection method. Section V discusses the experimental results and performance evaluation. Finally, Section VI concludes this paper.

II. Related Works

A. Dynamic Split Learning with Resource-Aware Partitioning

Early work on adaptive split learning explored resource-aware partitioning, where the cut layer is selected based on a client's computational capability, memory availability, or processing delay. AdaptSFL represents a representative approach in this category, assigning shallower splits to weaker devices and deeper splits to more capable ones to mitigate straggler effects and balance training workload across heterogeneous clients[10]. By adjusting the split position according to device capacity, these methods improve training fairness and reduce round-level delays within federated or collaborative learning systems.

However, resource-driven strategies implicitly assume that computation is the dominant source of latency and thus do not account for fluctuations in communication performance. This becomes a critical limitation in vehicular edge environments, where RSUs often exhibit similar computational capability but operate under highly diverse wireless conditions caused by varying bandwidth, link distance, and interference. Because resource-aware schemes such as AdaptSFL select partitions without considering channel states, they remain inadequate for

VEI systems in which communication variability dominates end-to-end training delay. This limitation motivates approaches that rely on network-driven cut-selection, which forms the foundation of the ASFL method proposed in this study.

B. Communication-Aware Split Learning in Wireless Edge Networks

A second line of research focuses on communication-aware split learning, where the split position is chosen based on uplink latency, activation size, or estimated communication burden. EPSL is a notable example that jointly optimizes subchannel allocation, transmission power, and cut-layer selection to reduce per-round training delay in wireless edge networks[11]. By coordinating the communication resources and selecting split points that minimize activation-transfer overhead, such methods demonstrate improved responsiveness under constrained or congested wireless environments.

Despite their contributions, existing communication-focused approaches intertwine computation and communication optimization, making it difficult to isolate the effect of channel variability alone. Moreover, frameworks such as EPSL assume quasi-static or predictable channel conditions to enable large-scale optimization, an assumption that does not hold in dynamic vehicular environments where RSU–server link quality fluctuates rapidly. These schemes also do not explicitly address how Non-IID data imbalance across RSUs interacts with fluctuating communication conditions, reducing their robustness in VEI deployments. These limitations highlight the need for a lightweight, channel-driven split-selection mechanism, which is the key objective of the ASFL framework proposed in this work.

III. ARCHITECTURE OVERVIEW

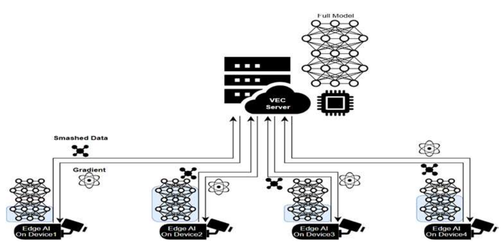

Fig 2. System Model

The overall learning architecture is illustrated in Fig 2, where multiple RSU-mounted edge devices collaborate with a centralized VEC server through a split neural network structure. Each edge device executes the front portion of the model and transmits its resulting smashed data to the server, while the server completes the remaining layers and returns gradients for local parameter updates. This RSU–server co-learning mechanism enables low-latency perception at the edge while leveraging the server's computational capacity for deeper feature extraction and global aggregation. The architecture explicitly accommodates heterogeneous RSU communication conditions, allowing the split position to be adjusted dynamically according to channel quality in the proposed ASFL framework. The remainder of this section describes the system model, communication assumptions, and operational workflow that form the foundation of the proposed ASFL methodology.

A. System Model

We consider a vehicular edge intelligence (VEI) system composed of four RSU-mounted edge devices and a centralized VEC server that collaboratively train a deep neural network under a split learning paradigm. Each RSU is equipped with a camera that continuously captures road-scene images within its coverage area and executes the front portion of a shared convolutional backbone. The server is responsible for processing the remaining layers, aggregating updates across RSUs, and maintaining the global model state.

The end-to-end network is based on a ResNet-18 architecture and is decomposed into an RSU-side subnetwork and a server-side subnetwork. This decomposition enables computation to be distributed across RSUs and the server while avoiding the exchange of raw visual data. The overall model is defined as

$$f(x; W) = f_{\mathrm{SRV}}(f_{\mathrm{RSU}}(x; W^{\mathrm{RSU}});\ W^{\mathrm{SRV}}) \qquad (1)$$

where W^{RSU} and W^{SRV} denote the parameters of the RSU-side and server-side subnetworks, respectively.

To support adaptive model partitioning, the RSU-side subnetwork provides five candidate split positions, denoted as $L = 0,1,2,3,4$, corresponding to boundaries between major residual blocks of the ResNet-18 backbone. Smaller values of L assign only shallow layers to the RSU, resulting in compact intermediate representations and lower communication overhead, whereas larger values of L place deeper layers at the RSU, producing more expressive but larger activation tensors.

For a given input image x, RSU i performs partial forward propagation up to the selected split layer L, yielding an intermediate activation

$$h_{i,L} = f_{\mathrm{RSU}}^{(L)}(x) \qquad (2)$$

This activation, referred to as smashed data, is transmitted to the server, which completes the remaining forward pass, computes the training loss $\mathcal{L}_i$, and derives the corresponding gradient.

Due to the hierarchical structure of ResNet-18, the size of the smashed data varies depending on the selected cut layer. As the cut depth increases, the spatial resolution of feature maps is progressively reduced while the number of channels increases, resulting in different activation sizes across split positions. Consequently, the communication overhead between RSUs and the server is directly affected by the chosen cut layer through the size of the transmitted smashed data.

$$\nabla W_{i,L}^{\mathrm{RSU}} = \frac{\partial \mathcal{L}_i}{\partial W_{i,L}^{\mathrm{RSU}}} \qquad (3)$$

The gradient is then returned to the RSU for local parameter updates.

All RSUs are assumed to possess comparable computational capabilities, such that performance heterogeneity in the considered system arises primarily from communication-related factors rather than device-side processing power. This modeling choice is intentionally adopted to isolate the impact of network variability on the split selection process and to focus on communication-driven adaptation in infrastructure-powered RSU environments. By controlling computational heterogeneity, the proposed framework can explicitly analyze how time-varying network conditions influence distributed learning behavior, which forms the basis for the adaptive split selection strategy described in subsequent sections.

B. Communication Model

The communication model characterizes the data transmission process between RSU-mounted edge devices and the centralized VEC server during split federated learning. Since RSUs and the server are connected through wireless backhaul links, the end-to-end training latency is strongly influenced by channel conditions such as available bandwidth, propagation distance, and interference. In the considered system, communication delay arises primarily from the transmission of smashed data from RSUs to the server and the delivery of gradient information in the reverse direction.

Let B_i denote the available uplink bandwidth between RSU $i$and the server. When RSU $i$selects split position L, the size of the corresponding smashed activation $h_{i,L}$depends on the depth of the RSU-side subnetwork. The uplink transmission latency is therefore modeled as

$$T_i^{\mathrm{up}}(L) = \frac{|\, h_{i,L} \,|}{B_i} \tag{4}$$

where $|\, h_{i,L} \,|$represents the data volume of the smashed activation.

Similarly, after completing the server-side backward pass, the gradient associated with the RSU-side parameters is transmitted back to RSU i. The downlink latency is given by

$$T_i^{\mathrm{down}}(L) = \frac{|\, \nabla W_{i,L}^{\mathrm{RSU}} \,|}{B_i} \tag{5}$$

where $|\, \nabla W_{i,L}^{\mathrm{RSU}} \,|$denotes the size of the returned gradient information. Because gradient size varies marginally across split positions compared to smashed data, uplink latency dominates the overall communication cost.

The total communication latency experienced by RSU $i$in a single training round is thus expressed as

$$T_i^{\mathrm{comm}}(L) = T_i^{\mathrm{up}}(L) + T_i^{\mathrm{down}}(L) \tag{6}$$

Due to time-varying wireless conditions, the effective bandwidth $B_i$may fluctuate across RSUs and training rounds. As a result, a fixed split position cannot consistently balance communication efficiency and feature representation quality. This observation motivates a dynamic split selection strategy that adapts the partition point Lin response to real-time channel conditions, which is formally introduced in the proposed ASFL framework.

C. Split Federated Learning Workflow

The training process follows the Split Federated Learning (SplitFedV1) paradigm, in which multiple RSUs collaboratively train a shared model under the coordination of a centralized server. Training proceeds in synchronized rounds, and all RSUs participate in each round using their locally collected data.

At the beginning of round t, the VEC server broadcasts the current global model parameters $W^t$to all RSUs. Each RSU initializes its local RSU-side subnetwork with $W^{\mathrm{RSU},t}$ and performs forward propagation up to its selected split layer Lusing local input samples. The resulting smashed activation $h_{i,L}^t$is then transmitted to the server.

Upon receiving smashed data from participating RSUs, the server instantiates the corresponding server-side subnetworks and completes the forward and backward passes independently for each RSU. The server computes the local training loss $\mathcal{L}_i^t$and derives the gradients with respect to the RSU-side parameters. These gradients are subsequently transmitted back to each RSU, enabling local backward propagation and parameter updates.

After completing local updates, each RSU sends its updated RSU-side parameters $W_i^{\mathrm{RSU},t+1}$ to the server. The server aggregates the received parameters using a weighted averaging scheme consistent with federated learning, given by

$$W^{RSU,t+1} = \sum_{i=1}^{K} \frac{n_j}{\sum_{j=1}^{K} n_j} W_i^{RSU,t+1} \tag{7}$$

where $n_i$denotes the number of training samples held by RSU i. The aggregated parameters form the updated global model, which is redistributed to all RSUs at the start of the next training round.

The architecture described in this section establishes a split federated learning framework in which RSU-mounted edge devices and a centralized VEC server collaboratively train a shared model under heterogeneous communication conditions. While this structure enables distributed learning without exchanging raw data, it does not by itself address the performance degradation caused by fluctuating network quality when a fixed split layer is applied. To overcome this limitation, the following section formalizes a network-aware adaptation requirement and introduces an ASFL scheme that dynamically determines the split layer according to RSU–server channel conditions. This adaptive design aims to stabilize training and improve efficiency in vehicular edge environments with highly variable communication links.

To capture the communication cost associated with each split position, we model the uplink delay between RSU $i$and the server as a function of the selected cut layer. Let $\mathcal{S} = \{0,1,2,3,4\}$denote the predefined set of candidate cut layers, and let $S(L)$represent the size (in bits) of the smashed activations produced at cut layer $L \in \mathcal{S}$. Given the instantaneous uplink bandwidth B_i available to RSU i, the corresponding transmission delay is expressed as

IV. Proposed Scheme

In a split learning framework, the cut layer determines both the distribution of computational workload between RSUs and the server and the size of intermediate activations transmitted over the wireless backhaul. Assigning a deeper cut enables richer feature extraction at the RSU, which can be beneficial for learning under heterogeneous data distributions, but it also generates larger activation tensors that significantly increase uplink transmission cost. In contrast, a shallower cut reduces communication overhead by producing compact activations, while constraining the representational capacity of the RSU-side model. This trade-off becomes particularly critical in vehicular edge environments, where RSU–server channel conditions fluctuate over time and differ across deployment locations. Under such conditions, employing a fixed cut layer cannot consistently balance communication efficiency and learning effectiveness, motivating the need for a dynamic, network-aware split selection strategy.

$$T_i^{\text{tx}}(L) = \frac{S(L)}{B_i} \tag{8}$$

This formulation reflects the fact that deeper cuts, which place more layers at the RSU, generate larger activations and thus incur longer transmission delays when bandwidth is limited.

Based on this model, the proposed method selects the cut layer for each RSU by minimizing the transmission delay under current channel conditions. For RSU i, the split layer is determined by

$$L_i^* = \arg\min_{L \in \mathcal{S}} \quad T_i^{\text{tx}}(L) \tag{9}$$

so that RSUs experiencing poor channel quality are assigned shallower cuts with smaller activation sizes, whereas RSUs with favorable channels can adopt deeper cuts to preserve richer feature representations. Because all RSUs are assumed to have comparable computational capabilities, this decision rule focuses adaptation solely on network variability rather than device-side processing differences.

Once each RSU has selected its cut layer and completed split forward and backward propagation with the server, RSU-side parameters are updated locally and then aggregated at the server following the SplitFedV1 protocol. Let $W_i^{(t)}$denote the RSU-side model parameters of RSU $i$after local updates in round t, and let $n_i$be the number of training samples used at RSU i. The server computes the updated global RSU-side model as

$$W^{(t+1)} = \sum_{i=1}^{K} \frac{n_j}{\sum_{j=1}^{K} n_j} W_i^{(t)} \tag{10}$$

This aggregation rule preserves the standard convergence behavior of split federated learning while allowing each RSU to employ a different cut layer according to its instantaneous channel state. As a result, the proposed adaptive scheme reduces unnecessary communication overhead for bandwidth-limited RSUs and improves the overall stability and efficiency of training under heterogeneous and time-varying vehicular network conditions.

Algorithm 1 Channel-Driven Adaptive Cut Selection

Require: Available uplink bandwidth B_i of RSU i, candidate cut set $\mathcal{S} = \{0, 1, 2, 3, 4\}$, activation size function $S(L)$
Ensure: Selected cut layer L_i^* for RSU i

```
Initialize T_min ← ∞
At the beginning of each training round:
for each L ∈ S do
    Estimate transmission delay T_i^tx(L) ← S(L)/B_i
    if T_i^tx(L) < T_min then
        T_min ← T_i^tx(L)
        L_i* ← L
    end if
end for
Select cut layer L_i* for split forward propagation
Transmit smashed data generated at layer L_i* to the server
return L_i*
```

V. Experimentation & Results

A. Simulation Environment

Table. 1. Simulation Parameters

Parameter	Description
Number of RSUs	4 RSU-mounted Edge Devices
Edge Server	Centralized VEC Server
Learning Framework	Proposed ASFL, Different cut selection ASFL
Training Model	ResNet-18
Training Rounds	200 rounds
Local Batch Size	32
Dataset	BDD100k(vehicle), Citypersons(pedestrian)

Table I presents the simulation parameters employed in this study. Four RSU-mounted edge devices collaboratively train a ResNet-18 model with a centralized VEC server under the proposed ASFL framework and its variants incorporating different cut-selection strategies. The training process is executed for 200 rounds with a fixed local batch size of 32, using vehicle perception data from the BDD100K dataset and pedestrian perception data from the CityPersons dataset.

B. Experimental Results

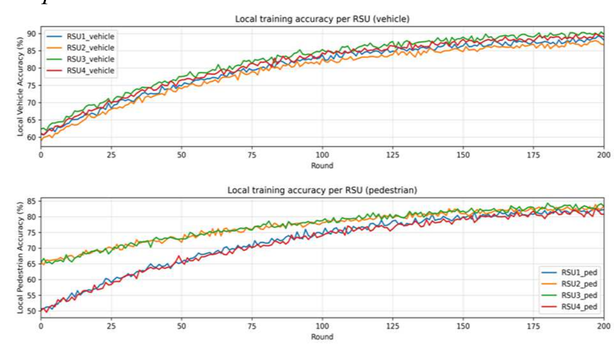

Fig 3. Local Training accuracy for each object

Fig. 3 illustrates the progression of local training accuracy for vehicle and pedestrian perception across the four RSUs. All RSUs show steady improvement and ultimately converge despite environmental differences. RSU 1 and 4, located in areas with minimal pedestrian, start with lower pedestrian-detection accuracy but gradually improve and approach the performance of RSU2 and RSU3, indicating that the collaborative learning process mitigates location-driven data imbalance.

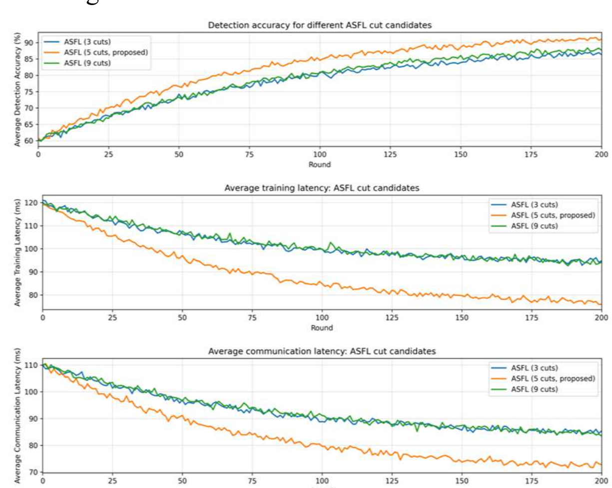

Fig 4. Accuracy and latency comparison with different cut layers

Fig. 4 compares the accuracy and latency characteristics of the ASFL framework under three different cut configurations. The 5-cut setting achieves the highest detection accuracy, indicating a more balanced partitioning of computation between RSUs and the VEC server. In contrast, the 3-cut and 9-cut settings show slower accuracy improvement due to their disproportionate local computation or communication demands. The training-latency results further highlight the efficiency of the proposed 5-cut approach, which converges to a substantially lower latency than the other two configurations. A similar trend is observed in communication latency, where the 5-cut scheme consistently maintains the smallest overhead. These results demonstrate that an appropriately chosen cut position can improve both model performance and resource utilization within the RSU–VEC collaborative learning environment.

Overall, the evaluation results demonstrate that the proposed ASFL framework provides consistent performance gains across heterogeneous RSU environments. The 5-cut configuration, in particular, achieves a more favorable balance between detection accuracy and delay, confirming the effectiveness of adaptive computation–communication partitioning in RSU–VEC collaborative learning. Furthermore, the convergence behavior observed across all RSUs, including those with limited pedestrian exposure, indicates that the framework effectively mitigates data imbalance and promotes stable model refinement. These findings collectively validate the robustness of the proposed approach and establish its suitability for deployment in practical multi-RSU edge intelligence systems.

VI. Conclusion

This paper presented a channel-driven ASFL framework for RSU–VEC collaborative perception, addressing the limitations of fixed split configurations under heterogeneous vehicular communication environments. By dynamically selecting the split layer based on RSU–server link conditions, the proposed approach effectively balances communication overhead and feature expressiveness, leading to improved detection accuracy and reduced training latency across RSUs with diverse sensing characteristics. Experimental results demonstrated that adaptive cut selection enables stable convergence even under non-IID data distributions, highlighting the benefit of network-aware partitioning in vehicular edge intelligence systems. While this work intentionally focuses on isolating the impact of network variability by assuming comparable computational capabilities among infrastructure-powered RSUs, future work will extend the proposed framework to jointly consider dynamic computational latency and communication conditions, enabling more comprehensive optimization under realistic, time-varying RSU workloads.

Acknowledgement

This work was supported by the Commercialization Promotion Agency for R&D Outcomes(COMPA) grant funded by the Korean Government(Ministry of Science and ICT). (RS-2023-00304695) This research was supported by the Regional Innovation System & Education(RISE) Glocal 30 program through the Daegu RISE Center, funded by the Ministry of Education(MOE) and the Daegu, Republic of Korea.(2025-RISE-03-001)

References

[1] T. Gong, L. Zhu, F. R. Yu, and T. Tang, "Edge intelligence in intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 9, pp. 8919–8944, 2023.

[2] X. Qiang, Z. Chang, C. Ye, T. Hamalainen, and G. Min, "Split federated learning empowered vehicular edge intelligence: Concept, adaptive design, and future directions," *IEEE Wireless Communications*, 2025.

[3] G. Cho, Y. Shinyama, J. Nakazato, K. Maruta, and K. Sakaguchi, "Object recognition network using continuous roadside cameras," in *2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring)*, IEEE, 2022, pp. 1–5.

[4] Z. Lu, H. Pan, Y. Dai, X. Si, and Y. Zhang, "Federated learning with non-iid data: A survey," *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 19188–19209, 2024.

[5] S. Moon and Y. Lim, "Split and federated learning with mobility in vehicular edge computing," in *2023 IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications (SERA)*, IEEE, 2023, pp. 35–38.

[6] J. Li, Y. Zhang, Y. Li, X. Gong, and W. Wang, "Fedsparse: A communication-efficient federated learning framework based on sparse updates," *Electronics*, vol. 13, no. 24, p. 5042, 2024.

[7] Z. Zhao, D. Liu, Y. Cao, T. Chen, S. Zhang, and H. Tang, "U-shaped Split Federated Learning: An Efficient Cross-Device Learning Framework with Enhanced Privacy-preserving," in *2023 9th International Conference on Computer and Communications (ICCC)*, IEEE, 2023, pp. 2182–2186.

[8] J. Dachille, C. Huang, and X. Liu, "The Impact of Cut Layer Selection in Split Federated Learning," *arXiv preprint arXiv:2412.15536*, 2024.

[9] C. F. Mecklenbrauker *et al.*, "Vehicular channel characterization and its implications for wireless system design and performance," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1189–1212, 2011.

[10] Z. Lin, G. Qu, W. Wei, X. Chen, and K. K. Leung, "Adaptsfl: Adaptive split federated learning in resource-constrained edge networks," *arXiv preprint arXiv:2403.13101*, 2024.

[11] Z. Lin *et al.*, "Efficient parallel split learning over resource-constrained wireless edge networks," *IEEE Transactions on Mobile Computing*, vol. 23, no. 10, pp. 9224–9239, 2024.